

**AMENDMENTS TO THE SPECIFICATION**

Please amend paragraph [0035] as follows:

[0035] Figure 2 is a diagram illustrating an example flow of information between a service consumer ("SC"), a service intermediary, and a service provider ("SP") in one embodiment. In this example, the service provider 203 does not provide its own sequence of codes. When the service consumer 201 wants to request services of a service provider, it selects a start code and generates and stores 1 a sequence ("S") of codes derived from that start code. In one embodiment, the number of codes in the sequence may be predefined, for example, in a contract ("K") between the service provider and the service consumer that is registered with the service intermediary 202. The service consumer then sends 2 a registration request to the service intermediary. The registration request may include the start code, end code, identity of the service provider, and identity of the contract under which the services are to be provided. The service intermediary may validate the request (e.g., ensure that the service provider and contract are valid) and stores 3 a registration record for the service consumer. The service intermediary may assign a unique registration number to each registration so that disputed charges can be tracked to the corresponding registration. The service intermediary then sends 4 a notification of the registration to the service provider. The notification may include the end code and the identity of the service consumer and the contract. The service provider may validate the notification and store 5 the information of the notification for use in verifying service requests. The service provider then responds 6 to the service intermediary confirming that it has accepted the notification of the registration. The service intermediary in turn responds 7 to the service consumer indicating that registration has been accepted. The service consumer then sends 8 a request to the service provider. Each request includes a code of the sequence. Upon receiving a request, the service provider retrieves 9 the last code that was previously received from the service consumer—initially retrieving the end code provided by the service intermediary. The service provider applies the one-way function to the received

code to determine whether it matches the retrieved code. If so, then the received code is verified as being correct and the service provider stores 10 the received code and provides the service. The service provider may send 11 the results of the performing the service to the service consumer. Alternatively, the service consumer may verify that the service provider performed the service in some other way. For example, if the service request was to send an authorization to a vending machine to dispense a product, then the user can visually confirm whether the service was provided. Steps 8-11 are repeated for each service that the service consumer requests, up to the predefined length of the sequence. Upon completion of all the services in the sequence, the service provider can charge the service consumer. The charge may include the unique identification of the registration. If the service consumer disputes the charge, the service provider can use the start code provided by the service consumer as non-repudiatable evidence that it provided the services. The service provider can provide the start code to the service intermediary as evidence. The service intermediary can compare it to the start code provided by the service consumer at registration to determine whether the service provider wins the dispute.

Please amend paragraph [0044] as follows:

**[0044]** Figure 10 is a block diagram illustrating components of the service consumer in one embodiment. The service consumer 1000 includes a service consumer code component 1001, a runtime component 1002, applications 1003 ("app1" ... "appn"), a code store 1004, a contract store 1005, and an application ("app") store 1006. The service consumer code component is responsible for generating sequences and registering the sequences with the service intermediary. The runtime component is responsible for providing an environment to the applications through which the applications can access the service providers. The runtime component ensures that an application that requests services in excess of its authorization is uninstalled and that a corresponding notification is sent to the service provider. The code store contains information relating to the registrations. Each entry identifies the service provider,

contract, sequence, and current index into the sequence for each registered sequence. The contract store contains information describing the contracts of the service consumer. Each entry in the contract store identifies the service provider and the contract terms. The application store contains information describing the limits of services for each application. Each entry of the application store may identify the application, a service provider, the authorized limit, and the current usage of that service.